Remarks

In the Office Action mailed September 15, 2004:

1. A new oath/declaration was required;
2. Claims 5, 12 and 15 were objected to, due to informalities;
3. Claims 1, 2, 7, 8, 9, 14, 15 and 19 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 5,202,975 (Rasbold), in view of U.S. Patent No. 5,901,147 (Joffe);
4. Claims 4, 5, 11, 12, 16, 17 and 20 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Rasbold, in view of U.S. Patent No. 5,377,336 (Eickmeyer); and
5. Claims 6, 13 and 18 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Rasbold, in view of U.S. Patent No. 5,941,983 (Gupta).

I. Oath/Declaration

Applicants request the need for a replacement oath/declaration be held in abeyance until allowable subject matter is identified.

II. Claim Informalities

Claims 5, 12 and 15 have been amended to correct the noted informalities.

III. Rasbold (U.S. Patent No. 5,202,975)

Rasbold is directed to a method for scheduling instructions for a processor, wherein the instructions are reordered in response to a simulation of a run-time environment (column 4, lines 3-7. In particular, after a Leader Set of instructions is assembled, instructions to place in a Ready Set are determined through a compile-time simulation (column 8, lines 19-21).

A Desired Issue Time (DIT) is computed for the Leader Set instructions, and instructions having a DIT less than the current (simulated) time are included in the Ready Set (column 8, lines 22-28). The DIT for an instruction is the latest time at which the instruction can be issued and still complete at the time it would have completed if it had been issued immediately (column 11, lines 21-24). Instructions in the Ready Set are then scheduled in order of their cost (column 11, lines 36-38).

Rasbold thus differs significantly from Applicants' invention.

A. Rasbold Schedules Instructions Based on Cost, Not on the Basis of Whether the Instructions Add or Remove Elements from a Memory Queue

As described above, after a Ready Set is populated with instructions in Rasbold, the instructions are scheduled in order of their cost. The cost is defined as "the execution time of the instruction plus the cumulative execution times of all other instructions which depend therefrom" (column 9, lines 36-39).

In contrast, in embodiments of the present invention, instructions are removed from a Ready Set based on their impact on a memory queue. In particular, the desire is to keep the memory queue filled as close to a threshold level as possible (page 7, lines 4-11). The threshold depends on the processor that services the queue; an example threshold is one entry less than full, in order to keep the processor from stalling.

Claimed embodiments of the invention (e.g., claims 1, 8) specify that if the memory queue is filled *above* a threshold, an instruction that would remove one or more elements from the queue is scheduled. Conversely, if the memory queue is filled to a level *below* the threshold, an instruction that would add one or more elements to the queue is scheduled.

Rasbold need not, and does not, determine what type of instruction it deals with and the instruction's effect on a memory queue. Rasbold simply schedules instructions based on their DIT, without regard to whether they increase or decrease the size of the memory queue. Thus, in Rasbold, even if the memory queue is relatively empty, instructions that remove more elements from the queue will continue to be scheduled if their DITs are less than the simulated time (column 4, lines 49-55). Conversely, even if the queue is relatively full, instructions that add yet more elements will continue to be scheduled if their DITs are less than the simulated time.

The Examiner relied upon Joffe (U.S. Patent No. 5,901,147) to inject into Rasbold the awareness of a queue threshold. However, simply *identifying* a queue threshold does not teach or suggest someone to place instructions into the queue based on how they will affect the queue when executed. As described below, despite addressing a threshold, Joffe merely indicates that exceeding that threshold may cause congestion – an issue unrelated to Applicants' invention – and so a threshold may be *changed.* This is far more passive than Applicants' proactive techniques for keeping a memory queue filled to a threshold.

B. Rasbold Does Not Schedule Instructions Heuristically

In an embodiment of the invention (e.g., claims 1, 8), an instruction in a Ready Set may be selected heuristically if an instruction is not chosen based on its effect on the memory queue. The portion of Rasbold cited against this merely specifies that instructions that cannot benefit from immediate issuance (because their DITs are distant) are *not* scheduled. This has the effect of causing those instructions with the closest DITs to be scheduled first. There is nothing heuristic in this selection process; it is mechanical. Instructions are removed from the Ready Set based on their DIT, not a heuristic algorithm.

IV. Joffe (U.S. Patent No. 5,901,147)

Joffe is directed to an apparatus and method for changing thresholds in ATM switches to avoid congestion (title).

A. Joffe Does Not Teach Scheduling Instructions Based on Queue Fullness

Joffe teaches that when a queue becomes too full, congestion may occur and, in an ATM switch, cells may be lost (column 1, lines 10-14). Therefore, Joffe teaches one to vary a queue's threshold based on congestion in the switch – when congestion increases, the threshold should be decreased (column 1, lines 37-41).

Joffe says nothing about choosing what to put in a queue to try to keep it filled to the threshold. In embodiments of the present invention, if the memory queue is filled *above* the threshold, then an instruction that would remove one or more elements from the queue is scheduled. Conversely, if the memory queue is filled to a level *below* the threshold, an instruction that would add one or more elements to the queue is scheduled.

V. Selected Claims

A. Claims 1-7

Regarding claim 1, as described above in Sections IV and V, neither Rasbold nor Joffe teach the queuing of members of a Ready Set of instructions depending on their impact on the queue. Rasbold merely teaches one to queue an instruction depending on its desired issue time, *not* whether it will add elements to or subtract elements from the queue.

In claim 2, it is more clearly stated that operation (i) of claim 1 relates to an instruction that will remove at least one element from the queue. The cited portion of Rasbold does not teach or suggest this idea. Instead, as the Examiner points out, Rasbold specifies that instructions that cannot benefit from immediate issuance should *remain* in the Leader Set – not be removed from it – while other instructions "bubble" to the top. The plain language of claim 2 requires choosing an instruction "that will remove at least one element from said first queue." Rasbold does not do this.

Similarly, claim 7 specifies "removing at least one element … from said first queue if said first queue is at least as full as said threshold." The cited portion of Rasbold may discuss delaying issuance of an instruction, but this *has no effect* on the memory queue, and certainly does not remove an element.

B. Claims 8-14

Regarding claim 8, as described in Sections IV and V, neither Rasbold nor Joffe teach the queuing of members of a Ready Set of instructions depending on their impact on the queue. Rasbold merely teaches one to queue an instruction depending on its desired issue time, *not* whether it will add elements to or subtract elements from the queue.

In claim 9, it is more clearly stated that operation (i) of claim 1 relates to an instruction that will remove at least one element from the queue. The cited portion of Rasbold does not teach or suggest this idea. Instead, as the Examiner points out, Rasbold specifies that instructions that cannot benefit from immediate issuance should *remain* in the Leader Set – not be removed from it – while other instructions "bubble" to the top. The plain language of claim 2 requires choosing an instruction "that will remove at least one element from said first queue." Rasbold does not do this.

Similarly, claim 14 specifies "removing at least one element … from said first queue if said first queue is at least as full as said threshold." The cited portion of Rasbold may discuss delaying issuance of an instruction, but this *has no effect* on the memory queue, and certainly does not remove an element.

C. Claims 19-20

Claim 19 includes the following elements not recited in claim 1:

(d1) selecting a node in said ready set that corresponds to an instruction that would generate a memory operation; and

(e1) selecting a node in said ready set that corresponds to an instruction that requires completion of a previous memory operation.

No portions of Rasbold or Joffe were cited against these elements.

Rasbold cannot teach these actions because, as described in Section IV, Rasbold removes instructions into a Ready Set based solely on their desired issue times. Rasbold need not, and does not, determine what type of instruction is involved, and whether the instruction will generate a new memory operation or complete a previous operation. Also as described above, Rasbold does not teach one to select nodes or instructions heuristically, but rather mechanically according to a single criterion.

Thus, claims 19-20 should be allowed.

CONCLUSION

No new matter has been added with the preceding amendments. It is submitted that the application is in suitable condition for allowance. Such action is respectfully requested. If prosecution of this application may be facilitated through a telephone interview, the Examiner is invited to contact Applicant's attorney identified below.

Respectfully submitted,

Date: October 19, 2004

By: Daniel E Vaughan 42,199
Daniel E. Vaughan (Registration No.)

Park, Vaughan & Fleming LLP
702 Marshall Street, Suite 310
Redwood City, CA 94063
(650) 474-1973: voice
(650) 474-1976: facsimile